### 项目名称

极简的音乐播放器开发

### 功能设计

网络接口获取音频

播放与暂停

切换下一首歌曲

播放音乐进度的可视化

自动播放下一首

### 软件简介

开发工具: python

相关第三方库:requests，PyQt5，qtawesome

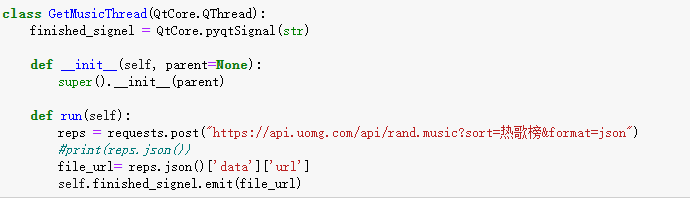
### 四、项目实现

**创建界面:**

通过pyqt进行界面的设计, 设置窗口大小及相关组件等, 包括了:播放状态标签, 播放/暂停按钮, 下一首歌曲按钮, 播放进度条, 程序关闭按钮。其中, 对界面顶部边框进行了隐藏, 并设置窗口背景透明度。

**获取歌曲:**

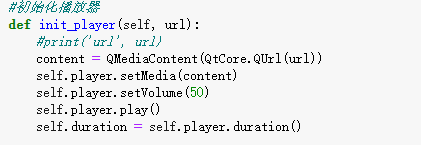
此次实验制作的播放器采用播放网络获取的音乐。通过请求网络接口随机返回一个音乐播放地址。用Qt的线程模块QThread进行包装, 作为一个异步的子线程调用。



**创建和链接信号槽:**

由于该播放器的边框为了美观隐藏起来, 因此通过创建的按钮作为程序的关闭, 主要过程为通过设置点击红点实现对程序关闭。

接着借助了Qt提供的QtMultimedia子模块的QMediaPlayer实现MP3文件的播放。

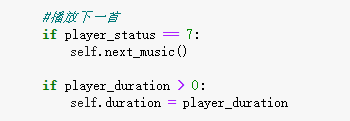


通过创建另一个方法对上一步创建的子线程进行调用, 完成信号连接到init\_player方法上实现对网络歌曲的播放



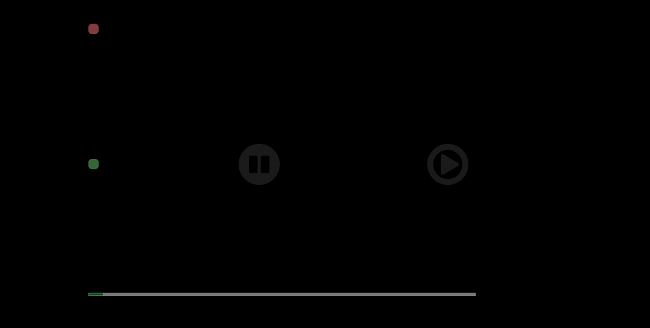
next\_music()方法调用GetMusicThread随机播放歌曲, 将歌曲url地址返回给init\_player()实现播放。其中, 播放按钮, 下一首按钮, 都可以调用他实现歌曲播放。

为了能够在播放完一首歌曲后自动获取并播放下一首, 创建了一个计时器, 通过间隔一定时间获取播放器状态, 若已经播放完音乐, 则调用next\_music播放下一首。(具体详情可见代码check\_music\_status)



**五**、**软件测试:**

下图为播放器界面:



（播放时）



（暂停时）

1.左上角红色按钮为关闭按钮，可关闭程序。

2.左边按钮为音乐状态标签，为绿色时播放器正常播放，为蓝色时播放暂停。

3.中间的按钮为播放/暂停，可控制播放与暂停。

4.右边的按钮为下一首按钮，可切换歌曲。

5.进度条显示正常。

6.进度条到达末尾，自动播放下一首。